



VEHICLE ROUTING PROBLEM WITH TIME WINDOWS PADA DISTRIBUTOR BAHAN MAKANAN

VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AT FOOD INGREDIENTS DISTRIBUTOR

Herry Christian Palit^{1, *)}, Sherly²⁾

*1) Industrial Engineering Department, Petra Christian University
Siwalankerto 121-131, Surabaya, 60236, Indonesia*

e-mail: herry@petra.ac.id

*2) Industrial Engineering Department, Petra Christian University
Siwalankerto 121-131, Surabaya, 60236, Indonesia*

ABSTRAK

Penelitian ini membahas penerapan *Vehicle Routing Problem with Time Windows* (VRPTW) pada perusahaan distributor bahan makanan. Selama ini penentuan rute pengiriman barang perusahaan berdasarkan pengelompokan area dari *customer* yang dituju dan kurang mempertimbangkan jalur rute secara keseluruhan, serta kapasitas dari kendaraan yang dipakai, sehingga biaya transportasi menjadi tinggi. Penelitian ini bertujuan untuk menentukan rute distribusi barang yang dapat meminimumkan biaya transportasi. Pada VRPTW ini menggunakan algoritma *tabu search* dan mempertimbangkan faktor kemacetan. Dari hasil simulasi VRPTW selama 5 hari, didapatkan bahwa terjadi penghematan rata-rata biaya transportasi sebesar 21,91%

Kata kunci: *Tabu Search, Vehicle Routing Problem with Time Windows*

ABSTRACT

This article discusses the application of Vehicle Routing Problem with Time Windows (VRPTW) at food ingredients distributor. During this time, the route of distribution based on customers' area clustering and do not consider the overall route also the capacity of the vehicle that make higher transportation cost. This article is aimed to find the route that minimize transportation cost. VRPTW uses tabu search algorithm and considers the traffic jam. The simulation of VRPTW for five days shows the average saving of transportation cost is around 21.91%.

Keywords: *Tabu Search, Vehicle Routing Problem with Time Windows*

PENDAHULUAN

Penelitian ini dilakukan pada sebuah distributor bahan makanan yang terletak di daerah Surabaya Timur, antara lain minyak goreng, mentega, *margarine*, keju, susu bubuk, dan coklat. Setiap kendaraan hanya keluar sekali dalam satu hari dari depot untuk mengirimkan barang ke *customer* dan setelah itu akan kembali ke depot setelah seluruh barang terkirim. Selama ini penentuan rute pengiriman barang berdasarkan pengelompokan area dari *customer* yang menjadi tujuannya dan kurang mempertimbangkan jalur rute secara keseluruhan, serta kapasitas dari kendaraan yang



dipakai, misalnya *customer* yang berada di daerah Surabaya Barat dikelompokkan menjadi satu kendaraan. Selain itu waktu yang tersedia untuk pengiriman barang mengikuti pihak *customer*. Perusahaan ingin menekan biaya transportasi yang terjadi saat ini. Oleh karena itu perusahaan ingin menentukan rancangan rute distribusi yang lebih baik dari 2 kendaraan yang dimilikinya, sehingga dapat meminimumkan biaya transportasi. Permasalahan ini termasuk dalam kasus *Vehicle Routing Problem with Time Windows* (VRPTW).

Vehicle Routing Problem (VRP) merupakan pemecahan masalah untuk menentukan rute kendaraan yang melayani beberapa pelanggan, dengan kapasitas angkut tertentu, dimana setiap pelanggan memiliki *demand*. Setiap pelanggan hanya boleh dikunjungi sekali dan total *demand* tidak boleh melebihi kapasitas angkut kendaraan yang dipakai. Selain itu, setiap kendaraan harus berangkat dan kembali pada *depot* yang sama. Tujuan dari VRP di sini adalah untuk meminimalkan total jarak tempuh dan jumlah armada yang digunakan. VRP merupakan variasi dari *m-TSP*, dimana terdapat *m-salesman* yang mengunjungi sejumlah kota dan tiap kota hanya satu salesman saja. Tiap *salesman* harus berasal dari *depot* dan berakhir di *depot* itu juga. Total jumlah *demand* pada suatu rute tidak boleh melebihi kapasitas dari armada yang dipakai untuk melewati rute tersebut. Tujuan dari VRP di sini adalah untuk meminimalkan total jarak tempuh dan jumlah armada yang digunakan. *Vehicle routing problem with time windows* merupakan perluasan dari permasalahan VRP. VRPTW memiliki batas tambahan yaitu sebuah jangka waktu tertentu. Jangka waktu tersebut merupakan jangka waktu pelayanan yang diinginkan pelanggan $[T_{ov}, T_{fv}]$, dimana T_{ov} dan T_{fv} merupakan *Time Windows* awal dan akhir. Berikut merupakan penentuan rute kendaraan dalam kasus VRPTW.

1. Setiap rute harus berawal dan berakhir pada *depot* yang sama.
2. Setiap *customer* hanya mengikuti satu rute.
3. Total beban dan durasi tidak boleh melebihi muatan kendaraan dan pembatas waktu.
4. Waktu pelayanan dimulai pada interval $[T_{ov}, T_{fv}]$, dan setiap *customer* meninggalkan *depot* dan kembali ke *depot* pada interval $[T_{ob}, T_{of}]$.
5. Total waktu perjalanan dari semua kendaraan diminimalisasi.

Berdasarkan penelitian-penelitian sebelumnya, VRPTW menggunakan beberapa metode *metaheuristik* untuk memecahkan masalah antara lain *ant colony*, *tabu search*, dan *simulated annealing*. Dalam penelitian kali ini akan digunakan algoritma *Tabu Search* karena *Tabu Search* dapat menemukan solusi yang mendekati optimal pada kasus penentuan rute (Cordeau *et al.*, 2001). Solusi optimal yang didapat memerlukan waktu penghitungan yang sangat lama dan memungkinkan terjadinya *cycling* (melakukan pertukaran yang sudah pernah dilakukan sehingga membuang waktu dan upaya penghitungan). Oleh karena itu, untuk menghindarinya dibuatlah sebuah struktur yang disebut dengan *tabu list*. *Tabu list* untuk menyimpan sekumpulan solusi yang baru dievaluasi. Pengujian empirik menghasilkan ukuran *tabu list* berkisar antara $0.33n$ dan $0.6n$ (n adalah jumlah *customer/job*) akan menghasilkan solusi yang baik, sedangkan pilihan terbaik untuk jumlah iterasi maksimum adalah $7n$ sampai $10n$ (Skorin-Kapov, 1995).

METODE

Algoritma *Tabu Search* bukan merupakan tipe *construction algorithm* melainkan tipe *improvement algorithm*, sehingga membutuhkan solusi awal. Solusi awal yang didapatkan dengan melihat jarak terpendek dari *node* asal. Setelah itu dari



solusi awal akan dihasilkan *optimal solution* dari solusi-solusi yang ada dengan menggunakan algoritma *Tabu Search*. Solusi awal yang terbentuk merupakan kondisi awal yang nantinya digunakan untuk tahap algoritma *Tabu Search*. Langkah pertama diawali dengan memasukkan beberapa data untuk diolah, sehingga didapatkan *initial solution*. Data-data yang di-input-kan adalah sebagai berikut.

n	= total customer
m	= total kendaraan yang akan dipakai
Q_k	= kapasitas muatan masing-masing kendaraan
v_k	= kecepatan rata-rata masing-masing kendaraan
k	= kendaraan ke $-k$, dimana $k = 1$
q_v	= jumlah permintaan barang dari masing-masing customer (kilogram)
s_v	= waktu pelayanan di masing-masing customer
i	= node asal kendaraan, dimana $i = 0, \dots, n$
j	= node tujuan kendaraan, dimana $j = 1, \dots, n$
d_{ij}	= jarak perjalanan dari i ke j
t_{ij}	= waktu yang dibutuhkan untuk perjalanan dari i ke j
T_{ov}	= waktu awal barang diperbolehkan untuk diterima
T_{fv}	= waktu akhir barang tidak diperbolehkan untuk diterima
$Time$	= waktu kendaraan di suatu node (i atau j)
Total jarak	= total jarak perjalanan kendaraan sampai di suatu node, dimana total jarak awal = 0 km
TP	= total waktu perjalanan kendaraan sampai di suatu node, dimana TP awal = 0 menit
BBM	= harga bahan bakar kendaraan
v	= customer ke v , dimana $v = 1, \dots, n$

Setelah memasukkan data-data tersebut, maka kendaraan mulai berangkat dan memilih customer yang memiliki jarak terdekat dari tempat asalnya, dan memiliki T_{fv} tercepat. Setiap kendaraan yang berangkat di suatu node akan dihitung waktunya ($Time_j$) dengan menambahkan waktu asal ($Time_i$) dengan waktu perjalanan dari asal (i) ke tujuan (j) yang dilambangkan dengan t_{ij} . Dalam menghitung t_{ij} , juga memperhitungkan kondisi kemacetan. Dalam kondisi macet, kendaraan akan berjalan dengan kecepatan rata-rata 20 km/jam, dengan prosentase 20%. Sehingga perhitungan t_{ij} adalah $\left(\frac{d_{ij}}{0,8v_k + 4} \right)$

Setelah menghitung $Time_j$, kemudian diperiksa apakah $Time_j$ lebih kecil dari T_{ov} apabila lebih kecil otomatis akan menimbulkan waktu tunggu. Tetapi jika lebih besar atau melewati T_{ov} , cukup menghitung total waktu perjalanan (TP_j) dan $Time_j$ yang memperhitungkan waktu pelayanan (s_v) di masing-masing customer. Setiap pemberhentian di node tujuan, juga menghitung total jarak yang telah ditempuh oleh kendaraan. Saat menghitung *demand customer*, maka akan selalu di periksa apakah *demand customer* (q_v) lebih kecil dari kapasitas kendaraan yang dipakai (Q_k). Jika q_v lebih kecil, maka akan dihitung sisa dari kapasitas kendaraan setelah dikurangi dengan *demand customer*, dan kemudian kendaraan akan menuju ke v selanjutnya. Tetapi jika, q_v lebih besar dari kapasitas kendaraan, maka kendaraan akan kembali ke *depot* untuk mengganti kendaraan yang mungkin lebih besar. Saat kembali ke *depot*, maka dihitung kembali $Time_j$, TP_j , dan total jarak. Setelah v bertambah satu untuk melanjutkan perjalanan ke customer lainnya, harus diperiksa terlebih dahulu apakah v masih lebih kecil dari jumlah customer (n). Jika masih lebih kecil, maka kendaraan akan mencari customer yang terdekat. Sedangkan jika v telah melebihi n , maka kendaraan juga akan



kembali ke *depot*. Hal ini berarti bahwa kendaraan telah mendatangi semua *customer*. Kendaraan yang telah menemukan tujuan *customer* yang akan didatangi kemudian menghitung waktu tiba ($Time_j$) di *node* tersebut. Apabila $Time_j$ menunjukkan lebih besar dari T_{fv} , maka hal ini berarti kendaraan terlambat mengirim barang, maka kendaraan akan mencari kembali *customer* yang baru untuk didatangi. Tetapi jika $Time_j$ menunjukkan masih lebih kecil dari T_{fv} , maka kendaraan akan memeriksa apakah *demand customer* ini lebih kecil dari sisa kapasitas kendaraan. Jika tidak, maka kendaraan akan kembali ke *depot* untuk berganti kendaraan yang baru. Jika ya, maka rute ini akan termasuk dalam kendaraan yang sama. Kemudian mulai kembali ke tahap yang awal untuk memeriksa apakah $Time_j$ lebih kecil dari T_{ov} . Apabila masih lebih kecil kembali menghitung waktu tunggu, TP_j , dan $Time_j$. Apabila lebih besar, maka tidak perlu menghitung waktu tunggu. Hal ini terus berlanjut sampai total *customer* telah terpenuhi.

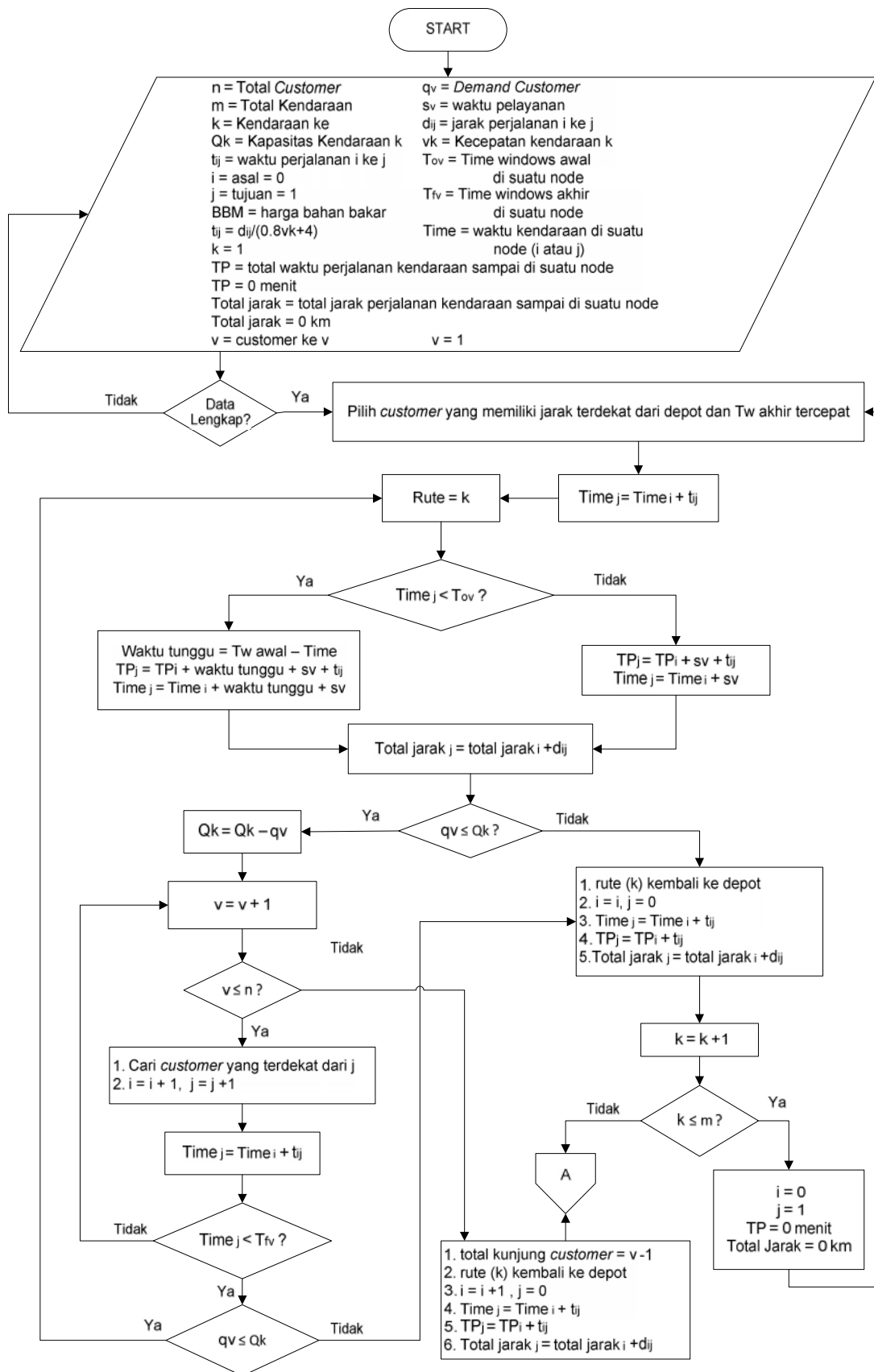
Jika total *customer* (n) telah terpenuhi, maka mulai mengurutkan rute dari kendaraan yang pertama dan menghitung biaya yang dikeluarkan untuk setiap kendaraan. Perhitungan biaya kendaraan didapat dari total jarak yang ditempuh dikalikan dengan harga BBM, tetapi dengan memperhitungkan unsur kemacetan. Menurut hasil wawancara dengan bagian logistik, kendaraan dalam kondisi macet memiliki perbandingan bahan bakar 1:3, sedangkan dalam kondisi normal memiliki perbandingan bahan bakar 1:8. Prosentase dalam sebuah perjalanan kendaraan akhir yaitu 80% kendaraan berjalan normal, sedangkan sisanya 20% kendaraan akan terjebak dalam kemacetan, sehingga perhitungan biaya per tiap kendaraan adalah $1/7$ dikalikan dengan total jarak. Rancangan solusi awal dalam kasus VRPTW dijelaskan lebih detail melalui *flowchart* yang ditunjukkan pada gambar 1, sedangkan untuk *flowchart* dari *tabu search* ditunjukkan pada gambar 2.

HASIL DAN PEMBAHASAN

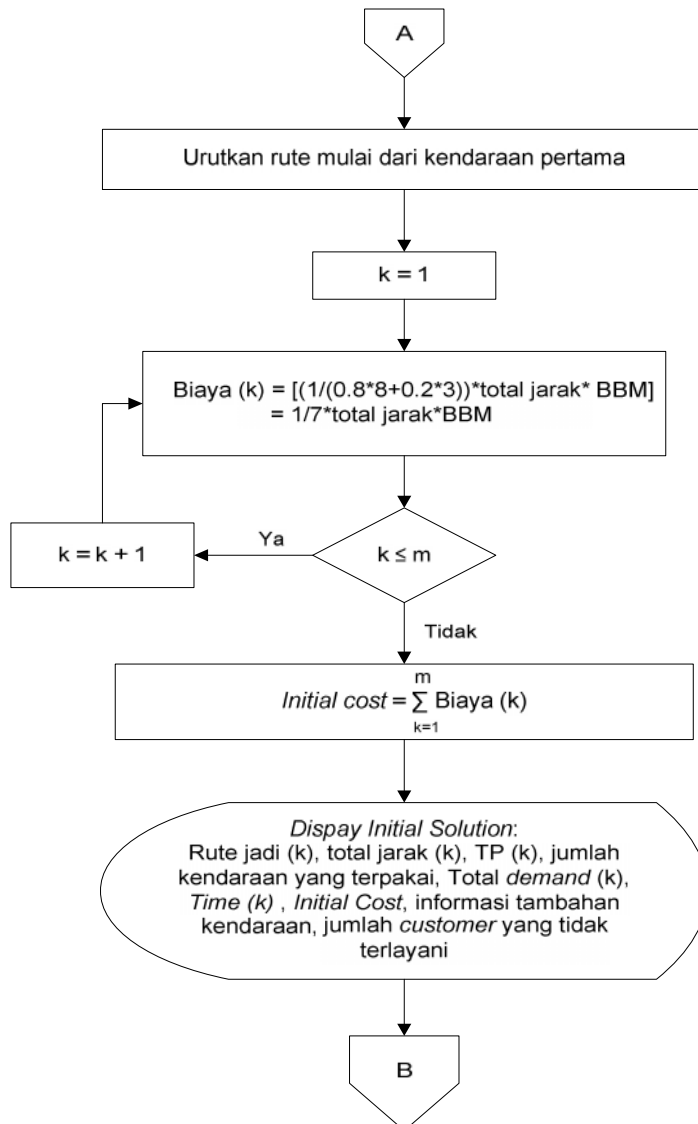
Perusahaan memiliki 2 kendaraan mobil *box* L300 dengan kapasitas 2,5 ton untuk pendistribusian barang. Penyelesaian kasus VRPTW ini dibuat dengan menggunakan program *Delphi*. Berdasarkan data permintaan *customer* selama 5 hari dengan replikasi sebanyak 10 kali, maka didapatkan perbandingan rute antara metode usulan dengan perusahaan beserta total biayanya seperti terlihat pada tabel 1, dimana terjadi penurunan rata-rata biaya transportasi untuk setiap harinya.

KESIMPULAN

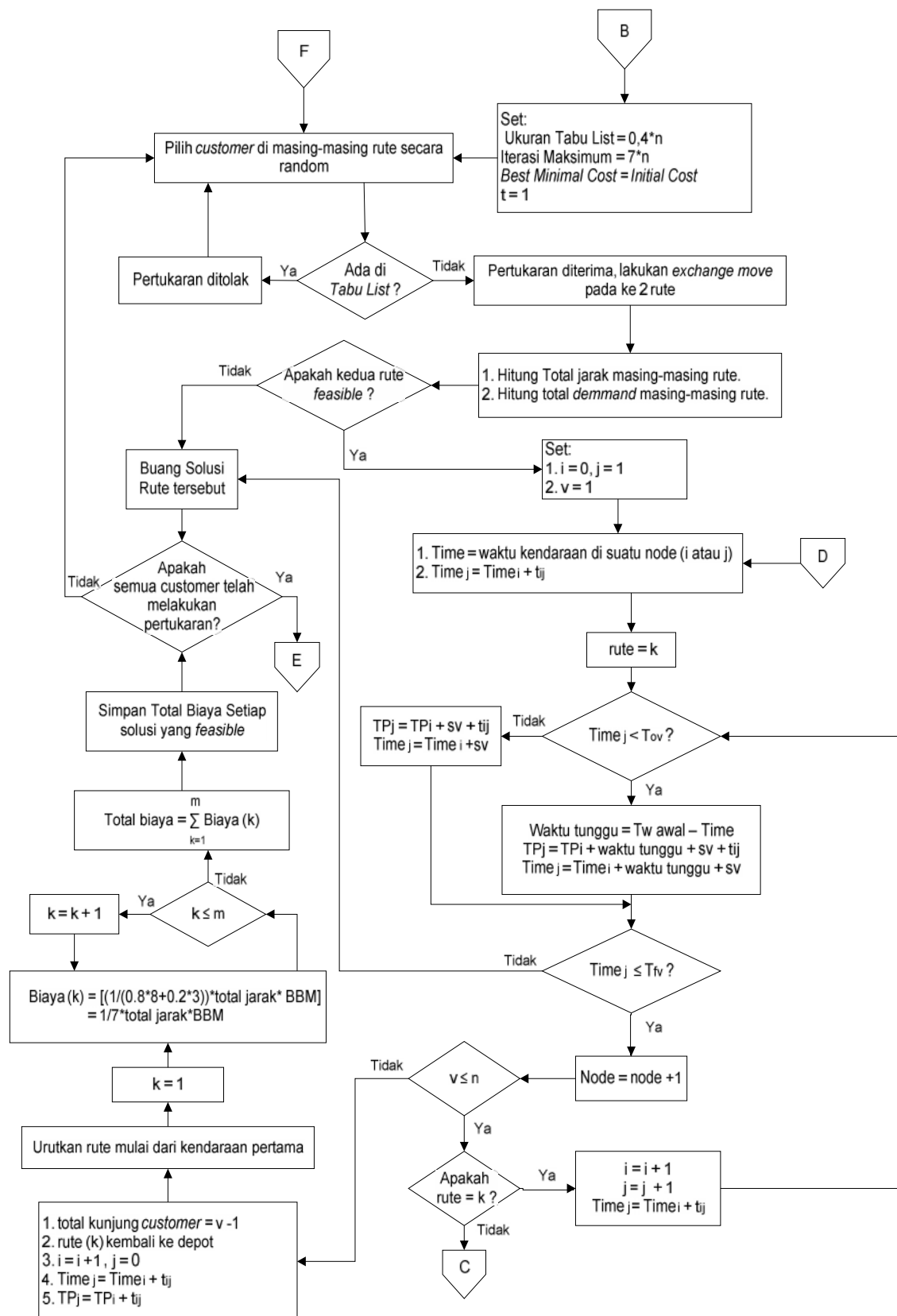
Penentuan rute pengiriman barang berdasarkan pengelompokan area *customer* yang dilakukan perusahaan selama ini mengakibatkan tingginya biaya transportasi. Melalui pendekatan VRPTW dengan menggunakan algoritma *tabu search* dapat menemukan solusi penentuan rute yang lebih baik. Hal ini dibuktikan melalui hasil simulasi usulan rancangan selama 5 hari, dimana didapatkan rata-rata prosentase penghematan biaya transportasi sebesar 21,91%.



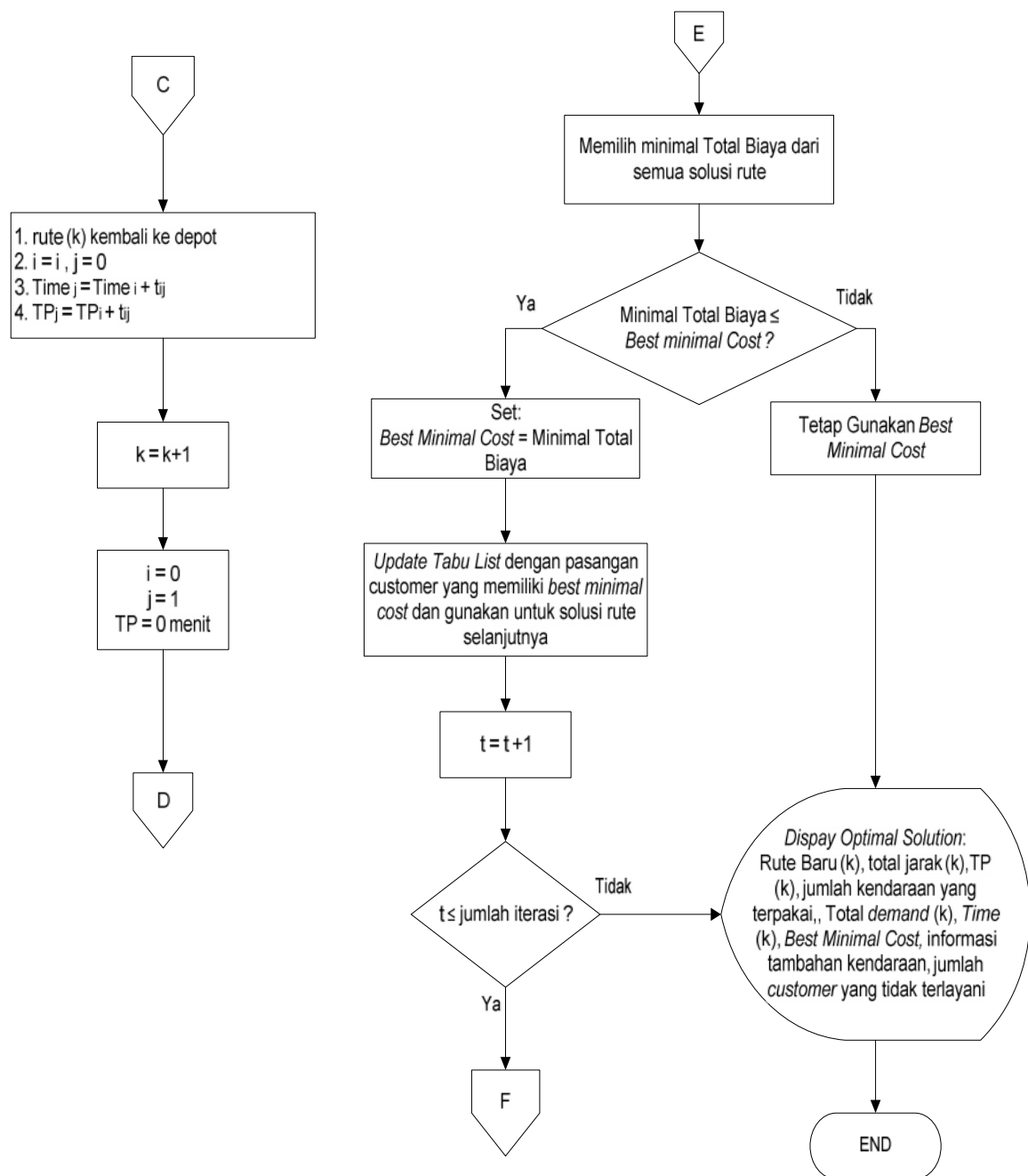
Gambar 1. Flowchart Algoritma Initial Solution



Gambar 2. Flowchart algoritma initial solution (sambungan)



Gambar 2. Flowchart Algoritma Tabu Search



Gambar 2. Flowchart Algoritma Tabu Search (lanjutan)



Tabel 1. Hasil VRPTW Dengan Algoritma Tabu Search

Hari	k	Rute Usulan	Total Jarak (km)	Total Biaya Usulan	Rute Perusahaan	Total Jarak (km)	Total Biaya Perusahaan
1	1	D-S20-S11-S1-S17-S18-S21-D	37	Rp 37.285,71	D-S7-S13-S18-S21-D	81	Rp 52.071,43
	2	D-S7-S13-S4-D	21		D-S1-S4-S11-S17-S20-D		
2	1	D-S20-S11-S16-S8-S12-S2-D	33	Rp 37.285,72	D-S2-S11-S12-S19-D	80	Rp 51.428,57
	2	D-S3-S19-S15-D	25		D-S3-S8-S15-S16-S20-D		
3	1	D-S5-S1-S17-S9-S18-D	32	Rp 38.571,43	D-S1-S14-S17-D	74	Rp 47.571,42
	2	D-S7-S6-S14-S10-D	28		D-S5-S6-S7-S9-S18-D		
4	1	D-S2-S8-S4-S12-S19-D	37	Rp 35.357,14	D-S2-S4-S8-S13-S22-D	66	Rp 42.428,57
	2	D-S7-S13-S22-D	18		D-S7-S12-S19-D		
5	1	D-S3-S16-S20-S1-S5-D	46	Rp 43.714,29	D-S5-S9-S14-S18-S21-D	83	Rp 53.357,14
	2	D-S18-S9-S14-S2-D	22		D-S1-S3-S16-S20-D		

DAFTAR PUSTAKA

Alonso F., Alvarez M. J. & Beasley J. E. (2008). A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of The Operational Research Society*, 59, 963-976.

Badeau P., Gendreau M., Guertin F., Potvin J.Y. & Tailard E. (1995). A parallel tabu search for the vehicle routing problem with time windows. Technical Report CRT 95-84. *Natural Sciences and Engineering Research Council of Canada*.

Cordeau, J.,F., Laporte, G. & Mercier A. (2001). A unified tabu search heruistic for vehicle routing problems with time windows. *Journal of The Operational Research Society*, 52, 928-936.

Gendreau Michael., Hertz Alain., & Laporte Gilbert. (1994) A tabu search heruistic for the vehicle routing problem. *Management Science*, 40, 10, 1276.

Lau H. C., Sim M. & Teo K.M. (2002). Vehicle routing problem with time windows and limited number of vehicles. *European Journal of Operational Research*, 148, 559-569.

Liong Choon.Yeun., Wan Rosmanira Ismail., Khairuddin Omar. & Zirour Mourad. (2008). Vehicle routing problem: model & solution. *Journal of Quality Measurement and Analysis*, 4(1), 205-218.

Skorin-Kapov J., (1995). Tabu search applied to the quadratic assigment problem. *Journal on Computing*. Vol.2, No1. , pp. 33-45.